

1 **Amendment to the Claims**

2 **In the Claims:**

3 Claims 1-20 were previously cancelled.

4 Please amend Claims 21, 30, and 36 as follows:

5 1. (Cancelled)

6 2. (Cancelled)

7 3. (Cancelled)

8 4. (Cancelled)

9 5. (Cancelled)

10 6. (Cancelled)

11 7. (Cancelled)

12 8. (Cancelled)

13 9. (Cancelled)

14 10. (Cancelled)

15 11. (Cancelled)

16 12. (Cancelled)

17 13. (Cancelled)

18 14. (Cancelled)

19 15. (Cancelled)

20 16. (Cancelled)

21 17. (Cancelled)

22 18. (Cancelled)

23 19. (Cancelled)

24 20. (Cancelled)

25 21. (Currently Amended) A method for accessing multiple types of electronic content,  
26 comprising:

27 receiving a request for a computer program to process an input to obtain an output  
28 comprising a type of content that is unknown to the computer program, wherein a service manager  
29 connects the computer program to at least one service container to process the input to obtain the  
30 output;

1 selecting at least one segment of computer code from a plurality of segments of  
2 computer code that will enable the computer program to process the input, when the at least one  
3 segment of computer code is executed along with the computer program, to provide the output  
4 comprising the type of content that is unknown to the computer program; and

5 executing the at least one segment of computer code along with the computer program  
6 to process the input and obtain the output comprising the type of content that is unknown to the  
7 computer program,

8 wherein the plurality of segments of computer code and the at least one segment of  
9 computer code are not executable as an independent computer program.

10 22. (Previously Presented) The method of claim 21, wherein selecting at least one segment  
11 of computer code comprises selecting at least two segments of computer code from the plurality of  
12 segments of computer code whose combined functionality will enable the computer program to  
13 process the input, when the at least two segments of computer code are executed along with the  
14 computer program, to provide the output comprising the type of content that is unknown to the  
15 computer program.

16 23. (Previously Presented) The method of claim 22, further comprising configuring the at  
17 least two segments of computer code to be executed along with the computer program in a particular  
18 order to provide a desired processing of the input.

19 24. (Previously Presented) The method of claim 22, further comprising configuring the at  
20 least two segments of computer code into a master-slave relationship that causes the execution of  
21 one of the at least two segments of computer code to be dependent on the execution of another of  
22 the at least two segments of computer code.

23 25. (Previously Presented) The method of claim 21, wherein executing the at least one  
24 segment of computer code comprises integrating the at least one segment of computer code into the  
25 computer program and executing the computer program to process the input and obtain the output  
26 comprising the type of content that is unknown to the computer program.

27 26. (Previously Presented) The method of claim 21, wherein receiving a request for a  
28 computer program to process an input comprises receiving a command to translate a word from a first  
29 language to a second language.  
30

1           27. (Previously Presented) The method of claim 21, wherein receiving a request for a  
2 computer program to process an input comprises receiving a command to convert a number from a  
3 first number format to a second number format.

4           28. (Previously Presented) The method of claim 21, wherein receiving a request for a  
5 computer program to process an input comprises receiving a command to convert a text object from  
6 a first text format to a second text format.

7           29. (Previously Presented) The method of claim 21, wherein receiving a request for a  
8 computer program to process an input comprises receiving a command to convert a graphical object  
9 from a first graphical format to a second graphical format.

10          30. (Currently Amended) A computer system for accessing multiple types of electronic  
11 content, comprising:

12               a processing unit;

13               a memory in communication with the processing unit; and

14               a computer program stored in the memory that provides instructions to the processing  
15 unit, wherein the processing unit is responsive to the instructions, operable for:

16                       identifying a plurality of segments of computer code that can be executed  
17 along with the computer program by the processing unit in response to the instructions;

18                       selecting, in response to an input command to access at least one type of  
19 content that the computer program is not configured to access, at least one segment of computer code  
20 from the plurality of segments of computer code that can be executed along with the computer  
21 program by the processing unit, in response to the instructions, to access the at least one type of  
22 content that the computer program is not configured to access, wherein a service manager connects  
23 the computer program to at least one service container in response to the input command; and

24                       executing the at least one segment of computer code along with the computer  
25 program to access the at least one type of content that the computer program is not configured to  
26 access;

27               wherein the plurality of segments of computer code and the at least one segment of  
28 computer code are not executable as an independent computer program.

29          31. (Previously Presented) The computer system of Claim 30, wherein the processing unit,  
30 responsive to the instructions, is further operable for:

1           arranging in the memory the at least one segment of computer code and a data,  
2 comprising the at least one type of content that the computer program is not configured to access, into  
3 a function-content group; and

4           interfacing the function-content group to the computer program to enable the computer  
5 program to access the at least one type of content that the computer program is not configured to  
6 access.

7           32. (Previously Presented) The computer system of Claim 30, wherein the processing unit,  
8 responsive to the instructions, is operable for identifying a plurality of segments of computer code by:

9           locating at least two segments of computer code from the plurality of segments of  
10 computer code that each comprise a portion of computer code that indicates they can be executed by  
11 the processing unit along with the computer program; and

12           generating a list in the memory comprising an identifier for each of the at least two  
13 segments of computer code that indicates that the at least two segments of computer code are  
14 available to be executed by the processing unit along with the computer program.

15           33. (Previously Presented) The computer system of Claim 30, wherein the processing unit,  
16 responsive to the instructions, is operable for selecting at least one segment of computer code by  
17 selecting at least two segments of computer code from the plurality of segments of computer code  
18 whose combined functionality will allow the computer program to access the at least one type of  
19 content that the computer program is not configured to access when the at least two segments of  
20 computer code are executed by the processing unit along with the computer program.

21           34. (Previously Presented) The computer system of claim 33, wherein the processing unit,  
22 responsive to the instructions, is further operable for configuring the at least two segments of  
23 computer code to be executed by the processing unit along with the computer program in a particular  
24 order to allow the computer program to access the at least one type of content that the computer  
25 program is not configured to access.

26           35. (Previously Presented) The computer system of claim 33, wherein the processing unit,  
27 responsive to the instructions, is further operable for configuring the at least two segments of  
28 computer code into a master-slave relationship that causes the execution of one of the at least two  
29 segments of computer code to be dependent on the execution of another of the at least two segments  
30 of computer code.

1           36. (Currently Amended) A computer-readable medium having computer-executable  
2 instructions for accessing multiple types of electronic content, comprising:

3                   logic for creating a list that comprises information about a plurality of segments of  
4 computer code that can be executed along with a computer program;

5                   logic for choosing at least one segment of computer code from the plurality of  
6 segments of computer code, based on the information in the list, that can be executed along with the  
7 computer program to process a type of data that the computer program is not designed to process;

8                   logic to execute the at least one segment of computer code along with the computer  
9 program in response to an input to provide an output of the type of data that the computer program is  
10 not designed to process, wherein a service manager connects the computer program to at least one  
11 service container in response to the input,

12                   wherein the plurality of segments of computer code and the at least one segment of  
13 computer code are not executable as an independent computer program.

14           37. (Previously Presented) The computer-readable medium of claim 36, further comprising  
15 logic for choosing at least two segments of computer code from the plurality of segments of computer  
16 code, based on the information in the list, which can be executed along with the computer program to  
17 process a type of data that the computer program is not designed to process.

18           38. (Previously Presented) The computer-readable medium of claim 37, further comprising  
19 logic for linking the at least two segments of computer code in a specific order of execution to  
20 provide a desired output of data that the computer program is not designed to process when the at  
21 least two segments of computer code are executed along with the computer program.

22           39. (Previously Presented) The computer-readable medium of claim 36, wherein the logic  
23 for creating a list that comprises information about a plurality of segments of computer code  
24 comprises:

25                   logic for identifying at least two segments of computer code that each comprise a  
26 registration code that indicates that they can be executed along with the computer program; and

27                   logic for generating a list comprising an identification code for each of the at least two  
28 segments of computer code that indicates that the at least two segments of computer code are  
29 available to be executed along with the computer program.

30 ///

1           40. (Previously Presented) The computer-readable medium of claim 36, further comprising  
2                   logic for arranging the at least one segment of computer code and a data element,  
3 comprising the type of data that the computer program is not designed to process, into a function-data  
4 group; and

5                   logic for interfacing the function-data group to the computer program to enable the  
6 computer program to provide the output of the type of data that the computer program is not designed  
7 to process.

8           Please add new Claims 41-50 as follows.

9           --41. (New) The method of Claim 21, wherein each service container comprises a data  
10 object, a code object, and a loader identification.

11           42. (New) The method of Claim 41, wherein each code object references at least one service  
12 object.

13           43. (New) The method of Claim 42, wherein each service object is stored in a cache,  
14 separate from each service container.

15           44. (New) The system of Claim 30, wherein each service container comprises a data object,  
16 a code object, and a loader identification.

17           45. (New) The system of Claim 44, wherein each code object references at least one service  
18 object.

19           46. (New) The system of Claim 45, wherein each service object is stored in a cache, separate  
20 from each service container.

21           47. (New) The computer-readable medium of Claim 36, wherein each service container  
22 comprises a data object, a code object, and a loader identification.

23           48. (New) The computer-readable medium of Claim 47, wherein each code object references  
24 at least one service object.

25           49. (New) The computer-readable medium of Claim 48, wherein each service object is  
26 stored in a cache, separate from each service container.

27           50. (New) A method for accessing multiple types of electronic content from a program,  
28 comprising the steps of:

29                   defining a plurality of service containers accessible to the program, each service  
30 container corresponding to a specific function, each service container including:

1                   a data object, such that each data object includes data required to enable the  
2 specific function corresponding to the service container to be achieved;  
3                   a code object, each code object referencing at least one segment of  
4 programming code stored separately from the service container, such that the at least one segment of  
5 programming code referenced by the code object includes programming code required to enable the  
6 specific function corresponding to the service container to be achieved; and  
7                   a loader identification, the loader identification providing the program with  
8 information required to load the segment of programming code referenced by the code object;  
9                   requesting an input to be processed to obtain an output;  
10                  identifying one of the plurality of service containers whose corresponding specific  
11 function is able to process the input;  
12                  parsing the service container thus identified, to determine:  
13                   data required to enable the specific function corresponding to the service  
14 container to be achieved;  
15                   an identity of the at least one segment of programming code required to enable  
16 the specific function corresponding to the service container to be achieved; and  
17                   information required to load each segment of programming code referenced by  
18 the code object;  
19                  retrieving the at least one segment of programming code required to enable the  
20 specific function corresponding to the service container to be achieved; and  
21                  executing the at least one segment of computer code under the control of the computer  
22 program to process the input and obtain the output.--  
23  
24  
25  
26  
27  
28  
29  
30